

Exam 2

Closed Book 3" x 5" notecard No Electronic Devices Closed Neighbor

**All cell phones and other electronic devices, including smart watches,
must be turned off and placed in your backpack**

Instructions

1. After we indicate you may start, print your uniqname at the top of every page of this booklet.
2. After we indicate you may start the exam, verify that you have received every page of the exam. There are 20 pages in this booklet, including this cover page.
3. You may have and reference one 3" x 5" notecard.
4. Assume all code and code fragments are syntactically valid, unless otherwise specified.
5. Assume/use only standard C++11.
6. In all the given code, if any character looks like a space, it is one.
7. Write clearly. If we cannot read your writing, it will be marked wrong. This includes indentation and curly braces.
8. Sign below and print your name and UMID. 10 pts off if we cannot read your name or UMID.

"I have neither given nor received aid on this examination, nor have I concealed any violations of the Honor Code."

SIGN your name above

Clearly and legibly print your **uniqname**

Clearly and legibly print your **full name (NOT uniqname)** above

Clearly and legibly print your **UMID** above

unqname: _____

Problem Number	Points Possible
Multiple Choice	84
Free Response 1	12
Free Response 2	22
Free Response 3	22
Free Response 4	22
Fill in the Blanks	18
	180

You can use the area below for working out problems.

Multiple Choice [4 points each]

Record your choices for all multiple choice questions using the circles next to each question. Fill

the single circle corresponding to your answer completely, like this: A B C D E

1. Which of the following about C++ classes is/are **true**?

 A B C D E

- A. Classes can contain data of different data types.
- B. Classes include both data and functionality.
- C. You can declare an array that holds elements of a class type.
- D. All of A, B, C are true.

2. Consider the following function declaration:

 A B C D E

`void printName(string &name);`

Which of the following code snippets contains valid function call(s)?

- A. `printName("EECS183");`
- B. `cout << printName("EECS 183");`
- C. `string name = "EECS 183";
printName(name);`
- D. `string name = "EECS 183";
cout << printName(name);`
- E. A and C both have valid function calls.

uniqname: _____

3. Which of the following about C++ arrays is false? 

- A B C D E

- A. Arrays are variables with one name that store a list of data items.
B. The size of an array cannot change at run time.
C. Each item of an array is individually accessible.
D. One array can have elements of different data types.

4. In C++, a class member function with the same name as the class would be described as which of the following?

- A B C D E

- A. Instance
B. Class
C. Constructor 
D. Member variable
E. main function

Consider the following code fragment, which will be **used in the next two questions**.

```
ifstream input;
input.open("book.txt");
string word;
int num = 0;
while (input >> word && input >> num) {
    cout << "Loop Body!";
}
input.close();
```

5. What prints if book.txt is opened successfully and contains: **Hello 183 Hello183**

- A B C D E

- A. Loop Body!
- B. Loop Body!Loop Body!
- C. Loop Body!Loop Body!Loop Body!Loop Body!
- D. eof
- E. Nothing prints

6. What prints if book.txt is opened successfully and contains: **183 183 183 183**

- A B C D E

- A. Loop Body!
- B. Loop Body!Loop Body!
- C. Loop Body!Loop Body!Loop Body!Loop Body!
- D. eof
- E. Nothing prints

uniqname: _____

Refer to the following **buggy** function definition which will be **used in the next two questions**

```
/**  
 * Requires: size > 0 and size <= number of elements of grid  
 * Modifies: nothing  
 * Effects: counts the number of capital 'A' characters  
 *           in grid for the first size number of elements  
 */  
int countAs(char grid[], int size) {  
    int count = 0;  
    for (int i = 0; i <= size; i++) {  
        if (grid[i] == 'A') {  
            count++;  
        }  
    }  
    return count;  
}
```

7. Given the following declaration, which call(s) to countAs would expose the bug?

```
char letters[4] = {'a', 'B', 'B', 'A'};
```

A B C D E

- A. cout << countAs(letters, 1);
- B. cout << countAs(letters, 2);
- C. cout << countAs(letters, 3);
- D. All of A, B, and C
- E. None of A, B, or C

8. Fixing the error in countAs can be accomplished by replacing the condition in the for loop with which of the following?

A B C D E

- A. i > size
- B. i = size
- C. i == size
- D. i != size
- E. Both A and C

Consider the following code, which will be used in the next three questions.

```
const int SIZE = 3;
char grid[SIZE][SIZE] = { {'0', 'X', '#'},
                         { 'X', '#', '#'},
                         { '#', '#', '#' } };

// print all the hashes(#) and none of the other characters
for (int row = 0; row < SIZE; row++) {
    for ( /**** what goes here? ****/ col < SIZE; col++ ) {
        cout << grid[row][col];
    }
}
```

9. What should go in the initialization of the inner for loop to print *only* the six hashes(#) in grid?

- A B C D E

- A. int col = 0;
- B. int col = SIZE;
- C. int col = row;
- D. int col = row - SIZE;
- E. int col = SIZE - row - 1;

10. Which of the following statement(s) would change the '0' to 'X' in grid?

- A B C D E

- A. grid(0, 0, 'X');
- B. grid = 'X';
- C. grid[0][0] = 'X';
- D. char grid[0][0] = 'X';
- E. Arrays can not be modified.

11. True (T) or False (F): the following snippet of code will print all the values in the array grid.

- T F

```
cout << grid << endl;
```

unique name: _____

12. Which of the following statements about C++ streams is true?

- A B C D E

- A. The only kind of streams are file streams.
- B. Streams can only be used for input.
- C. File streams can not be used with the insertion operator (`<<`).
- D. Streams can enter a fail state.
- E. Streams are never used for reading from files.

Consider the following Color class member function, which will be used in the next two questions.

```
/**  
 * Requires: redVal, greenVal, blueVal all in range [0,255]  
 * Modifies: red, green, blue.  
 * Effects: Constructor that sets RGB values.  
 */  
Color(int redVal, int greenVal, int blueVal);
```

13. Which of the following statements is an invalid way to use the Color class member function above?

- A B C D

- A. black(0, 0, 0);
- B. Color blue = Color(0, 0, 255);
- C. Color red = Color(200, 0, 0);
- D. Color black(0, 0, 0);

14. Which of the following about the Color class member function above is true?

- A B C D

- A. The function is a non-default constructor.
- B. The function has no parameters.
- C. The function will print an error message if redVal is out of range [0, 255].
- D. The function will modify Color class member variables.

1.

Consider the following class definition which will be **used in the next seven questions**.

```
const int DIMENSION = 100;

class Picture {
public:
    Picture();
    int countBlackPixels();
    void drawPixel(int row, int col, Color pixelColor);
    void drawRectangle(int startRow, int startCol,
                       int endRow, int endCol, Color pixelColor);
    void setName(string newName);
    string getName();

private:
    Color pixelData[DIMENSION][DIMENSION];
    string name;
};
```

15. Which of the following code snippets, in `main`, would create an instance of `Picture` with the name "Doge"

- | | | | | |
|-------------------------|-------------------------|-------------------------|------------------------------------|-------------------------|
| <input type="radio"/> A | <input type="radio"/> B | <input type="radio"/> C | <input checked="" type="radio"/> D | <input type="radio"/> E |
|-------------------------|-------------------------|-------------------------|------------------------------------|-------------------------|

- A. Picture myPic;
- B. Picture myPic("Doge");
- C. Picture myPic();
myPic.setName("Doge");
- D. Picture myPic;
myPic.setName("Doge");
- E. Picture myPic = Picture();
myPic.name = "Doge";

unique name: _____

16. Which of the following about the Picture class member drawRectangle is/are true?

(A) (B) (C) (D) **(E)**

- A. drawRectangle is a default constructor. **X**
- B. drawRectangle is a non-default constructor. **X**
- C. drawRectangle has no parameters. **X**
- D. drawRectangle has no return type. **X**
- E. drawRectangle can only be called using a declared instance of the Picture class. **C**

17. Within the definition of getName function, what is the correct statement to return the name private data member?

(A) (B) **(C)** **(D)** (E)

- A. cout << name;
- B. return "name";
- C. **return name;**
- D. cout << "Doge";
- E. return getName();

18. Consider the following variable declaration:

Picture topPic;

int

Which of the following is a valid call to the countBlackPixels function?

(A) (B) (C) **(D)** (E)

- A. countBlackPixels(**topPic**); **X**
- B. cout << countBlackPixels(**topPic**); **X**
- C. int black = Picture.**countBlackPixels()**;
- D. int black = topPic.countBlackPixels(); **C**
- E. int black = topPic.countBlackPixels; **X**

19. Which of the following creates an array of Picture instances named pics?

-
- A
-
- B
-
- C
-
- D
-
- E

- A. Picture pics;
 B. Picture pics[5];
 C. Picture[5];
 D. Picture([5]);
 E. Picture[5] pics();

20. Consider the array declaration of type Picture from the previous problem (#19). How would you print the name member variable of the first element of the array pics?

-
- A
-
- B
-
- C
-
- D
-
- E

- A. cout << pics[i].name;
 B. cout << pics[0].getName();
 C. cout << pics.getName();
 D. pics.printName();
 E. pics[0].printName();

21. Which of the following is a valid way to modify the data member pixelData given the following code snippet?

```
Color black;
Picture newPic;
```

-
- A
-
- B
-
- C
-
- D
-
- E

- A. newPic.~~pixelData~~[0][0] = black;
 B. newPic.~~pixelData~~(black);
 C. drawPixel(newPic, black);
 D. newPic.drawPixel(black);
 E. newPic.drawPixel(0, 0, black);

uniqname: _____

Page left intentionally blank.

You can use the area below for working out problems.

The following class definition in the Color.h file is relevant to the remaining questions on the exam.

```
class Color
{
public:
    /**
     * Requires: Nothing.
     * Modifies: red, green, blue.
     * Effects: Default constructor. Sets color to black (0,0,0).
    */
    Color();

    /**
     * Requires: redVal, greenVal, blueVal all in range [0,255]
     * Modifies: red, green, blue.
     * Effects: Constructor that sets RGB values.
    */
    Color(int redVal, int greenVal, int blueVal);

    /**
     * Requires: Nothing.
     * Modifies: Nothing.
     * Effects: Returns true if red, green, and blue
     *          all have a value of 0.
     *          Returns false otherwise.
    */
    bool isBlack();

private:
    int red;
    int green;
    int blue;
};
```

The following class definition in the Picture.h file is relevant to the remaining questions on the exam.

```
#include "Color.h"

const int DIMENSION = 100;

class Picture
{
public:
    /**
     * Requires: Nothing.
     * Modifies: pixelData
     * Effects: Default constructor.
     *          Sets all pixels to black, that is, Color(0, 0, 0)
    */
    Picture();
```

uniqueName: _____

```
/*
 * Requires: Nothing.
 * Modifies: Nothing.
 * Effects: Counts the number of pixels in pixelData
 *           that are black, i.e., Color(0, 0, 0)
 */
int countBlackPixels();

/**
 * Requires: Nothing.
 * Modifies: pixelData, cout
 * Effects: If row and col are within the bounds of array pixelData,
 *           i.e., between 0 and DIMENSION - 1, inclusive, sets the
 *           element of pixelData at [row][col] to pixelColor.
 *           Otherwise does not modify pixelData and prints the
 *           message "Out of bounds"
 */
void drawPixel(int row, int col, Color pixelColor);

/**
 * Requires: {startRow, startCol, endRow, endCol}
 *           all between 0 to DIMENSION - 1, inclusive, i.e., they
 *           are all valid indices of pixelData
 *           endRow >= startRow && endCol >= startCol
 * Modifies: pixelData
 * Effects: Draws a solid block of pixels pixelColor
 *           in a rectangle defined by the points [startRow][startCol]
 *           to [endRow][endCol] in pixelData
 */
void drawRectangle(int startRow, int startCol,
                   int endRow, int endCol, Color pixelColor);

/**
 * Requires: Nothing.
 * Modifies: name
 * Effects: Sets name to newName
 */
void setName(string newName);

/**
 * Requires: Nothing.
 * Modifies: Nothing.
 * Effects: Returns name
 */
string getName();

private:
    Color pixelData[DIMENSION][DIMENSION];
    string name;
};
```

Free Response 1 [12 points]

Fill in the boxes and write the definition of the **isBlack** member function from the Color class in the box below.

Note: assume your definition is **inside of Color.cpp**, which #includes Color.h.

```
/**  
 * Requires: Nothing.  
 * Modifies: Nothing.  
 * Effects: Returns true if red, green, and blue  
 *           all have a value of 0.  
 *           Returns false otherwise.  
 */
```

```
bool Color::isBlack() {  
  
    if (red == 0 && green == 0 && blue == 0) {  
        return true;  
    }  
  
    else {  
        return false;  
    }  
  
}
```

uniqueName: _____

Free Response 2 [22 points]

Fill in the boxes and write the definition of the **countBlackPixels** member function from the Picture class in the box below.

Note: assume your definition is **inside of Picture.cpp**, which **#includes Picture.h**.
Your solution **must** use the **Color** class member function **isBlack**.

```
/**  
 * Requires: Nothing.  
 * Modifies: Nothing.  
 * Effects: Returns the number of pixels in pixelData  
 *           that are black, i.e., Color(0, 0, 0)  
 */
```

```
int Picture::countBlackPixels() {  
  
    int number = 0;  
    for (int i = 0; i < DIMENSION; i++) {  
        for (int j = 0; j < DIMENSION; j++) {  
            if (pixelData[i][j].isBlack())  
                number++;  
        }  
    }  
    return number;  
}
```

Free Response 3 [22 points]

Fill in the boxes and write the definition of the **drawPixel** member function from the Picture class in the box below.

Note: assume your definition is **inside of Picture.cpp**, which #includes Picture.h.

```
/*
 * Requires: Nothing.
 * Modifies: pixelData, cout
 * Effects: If row and col are within the bounds of array pixelData,
 *           i.e., between 0 and DIMENSION - 1, inclusive, sets the
 *           element of pixelData at [row][col] to pixelColor.
 *           Otherwise does not modify pixelColor and prints the
 *           message "Out of bounds"
 */

```

Void Picture::drawPixel

```
if (row < 0 || row > DIMENSION - 1 || col < 0 || col > 99) {
    cout << "Out of bounds" << endl;
}
else {
    pixelColor = pixelData[row][col];
}
```

uniqueName: _____

Free Response 4 [22 points]

Fill in the boxes and write the definition of the `drawRectangle` member function from the Picture class in the box below.

Note: assume your definition is inside of `Picture.cpp`, which `#includes Picture.h`. Your solution **must** use the Picture class member function `drawPixel`.

```
/**  
 * Requires: {startRow, startCol, endRow, endCol}  
 *             all between 0 and DIMENSION -1, inclusive  
 *             endRow >= startRow && endCol >= startCol  
 * Modifies: pixelData  
 * Effects: Draws a filled in block of pixels pixelColor  
 *           in a rectangle whose top left corner is  
 *           [startRow][startCol] and bottom right corner  
 *           [endRow][endCol] in pixelData  
 */
```

Void Picture:: drawRectangle

```
(int startRow,  
 int startCol, int endRow, int endCol, Color pixelColor) {
```

```
for (int i = startRow; i <= endRow; i++) {  
    for (int j = startCol; j <= endCol; j++) {  
        drawPixel (i, j, pixelColor);  
    }  
}
```

}

Fill In The Blank [18 points] - Fill in the blanks in the following program.

```
#include <iostream>
#include <string>
#include "Color.h"
// fill in file dependency
#include "Picture.h"
using namespace std;
int main() {
    Picture newPic;
    newPic.setName("newPicture");
    // call countBlackPixels
    cout << "Number of black pixels: " <<
        newPic.countBlackPixels() << endl;
    // create any valid color
    Color myColor(200, 201, 202);
    // call drawPixel, draw a pixel at row = 40, col = 50
    newPic.drawPixel(40, 50, myColor);
    // call drawRectangle, draw a rectangle
    // starting at row = 40, col = 50 through row = 60, col = 70
    newPic.drawRectangle(40, 50, 60, 70,
    myColor);
    return 0;
}
```

uniqname: _____

Page left intentionally blank.

You can use the area below for working out problems.